

Amendments to the Specification:

Please replace the paragraph on page 1, lines 23-29 with the following amended paragraph:

a1
As near instant access to information through the information-browser model becomes a mainstay of global commerce, increased attention is being focused on the usability and clarity of purpose of online-content as it is displayed to a user. Being able to record the visual stimuli that is being presented to the user at any given time enables the study of how users react to that stimuli. Accordingly, the ability to accurately and efficiently record online-content is becoming a critical goal [the a] of the content design process.

Please replace the paragraph on page 4, lines 10-20 with the following amended paragraph:

a2
In one embodiment, the invention is a device for interpreting specific events as reported from a browser interface. The events are recorded in a selected format and syntax to form a primary index of events and related outcomes which comprise the user's interface experience. In operation, the device[s] detects events as detected at the browser interface. Next, the device[s] identifies, categorizes, and filters detected events as to their relevance to the visual stimuli being presented to the user. The device[s] then assigns a unique identifier, preferably in the form of an alphanumeric string, to each relevant event. Furthermore, the device[s] records selected events into a log (along with their categorization), a unique identifier, and the time in which the event occurred.

Please replace the paragraph on page 5, lines 1-9 with the following amended paragraph:

a³ Yet another embodiment is a device that records changes in parameters that may impact the visual stimuli. The device performs readings of several parameters, including the horizontal position, vertical position, and size of each instance of a browser window currently being displayed, along with the network address, horizontal offset, and vertical offset of online-content current being displayed within each browser window. Next, the device records these parameters in a log file in such a manner that they are related to the original event, preferably by using a[n] unique identifier associated with the original event.

Please replace the paragraph on page 18, lines 22-30 to page 19, lines 1-7 with the following paragraph:

a⁴ Fig. 3c illustrates an alternative embodiment of a relational block diagram in which a control application 372 is configured to work with a browser-interface 200 to record changes in the visual stimuli being displayed to a user 115 as the user 115 interacts with online-content in web pages 381, 382, and 384. As the user requests information from a host computer 130 via a computer network 120, the host computer 130 responds by sending individual documents 380 of online-content. As these documents 380 are being received and loaded by the browser-interface 374, the control program initiates a process block 398 that appends a messenger applet 350 to each document 380 thereby creating a new (modified) version of the document 352 containing the appended messenger applet 354. As these new documents are instantiated in web pages 381, 382, and 384, the messenger applet 356, 357, and 358 are used to monitor events which may cause changes in the visual stimuli being displayed to the user and communicate related data to

a⁴
another aspect of the control application on 372 that receives these messages 396, and writes this data to a storage device 140, and/or an external process 397.

[Please replace the paragraph on page 19, lines 8-18 with the following amended paragraph:]

One skilled in the art will understand that there are many methods for appending information to a document 380 as it is loaded by a browser-interface 374 depending on both the context in which the browser-interface is being utilized and browser dependent interface options. In a preferred embodiment of an Internet WWW browser manufactured by Microsoft Corporation® hosted on a computer running the Microsoft Windows® operating system, a JavaScript function 398 is used to call the DHTML method "CreateElement()" upon the DocumentComplete event as reported by the browser-interface. In this embodiment, the JavaScript function 398 is used to embed [a] the messenger applet 350 in the form of an appended messenger applet 354 (such as an Active X Control) [ActiveX control 354], which is appended to each document of online-content 352.

Please replace the paragraph on page 19, lines 19-30 to page 20, lines 1-7 with the following amended paragraph:

a7
Once the browser-interface 374 instantiates each document 380 in [an] appropriate web pages 381, 384, and 382, the embedded messenger applets 356, 357, and 358 is then used to report these changes in its corresponding web pages 381, 384, and 382 to a receiving aspect 396 of the control application 372 that receives messages from these messenger applets 356, 357 and 358. In this preferred embodiment, any DHTML event handle including but not limited to, Window.OnScroll (when a user scrolls the online-content within the browser window) and Window.OnResize (when a user resizes the browser window) can be monitored by binding the DHTML event handle to functions of the messenger applets 356, 357 and 358. Upon each DHTML event, any number of properties can be recorded from the browser window, including but not limited to, Window.Location (the network address of the online-content being displayed within the browser window), Document.Body.ClientWidth (the width of the browser window), Document.Body.ClientHeight (the height of the browser window), Document.Body.ScrollWidth (the native width of the online-content being displayed within the browser window), and Document.Body.ScrollHeight (the native height of the online-content being displayed within the browser window).

Please replace the paragraph on page 22, lines 19-30 to the paragraph on page 23, lines 1-7 with the following amended paragraph:

Q5
FIG. 5 illustrates the method for determining the area of online-content 550, 560, and 570 that is displayed within a browser window 580 and 585 as it may change over time, as described by the content algorithm 500 of figure 5b. As illustrated, the example online-content 550 occupies an area of 260,000 pixels. First, in a measure act 510, the browser window 580 is positioned at the top of the online-content 560 with a two-dimensional offset of (0,0). In this position, the browser window 580 is shown to display the first 120,000 pixels of the online-content 560. In this immediate position, 140,000 pixels of online-content 560 are not displayed to a user. In a second measure act 520, the online-content 570 has been moved vertically within the browser window 585, giving a two-dimensional offset of (0,300). In this position, the browser window 585 is shown to display an area of 120,000 pixels of the online-content 570 located between the two-dimensional coordinates of (0,300), (400,300), (400,600), and (0,600). In this position, 140,000 pixels of online-content 570 are not displayed to a user appearing in two disparate areas with the first (a first hidden area 595) located between the two-dimensional coordinates of (0,0), (400,0), (0,299), and (400,299) and the second (a second hidden area [596] 590) located between the two-dimensional coordinates of (0,601), (400,601), (0,650) and (400,650).

Please replace the paragraph on page 30, lines 5-24 with the following amended paragraph:

ab The invention provides methods and devices that address problems encountered when attempting to accurately reconstruct visual stimuli being displayed to a user as they interact with online-content, typically through a browser interface. In one embodiment, the invention provides for the browser to maintain a record of selected technical parameters and relevant data that may impact the manner in which online-content is being displayed to the user, taking into consideration the current context in which the browser is being operated. In another embodiment, the invention is a device for recording events as reported from a browser interface. The events are recorded in a selected format and syntax to form a primary index of events and related outcomes which comprise the user's interface experience. In operation, the device[s] detects events as detected at the browser interface. Next, the device[s] identifies, categorizes, and filters detected events as to their relevance to the visual stimuli being presented to the user. The device[s] then assigns a unique identifier, preferably in the form of an alphanumeric string, to each relevant event. Furthermore, the device[s] records selected events into a log (along with their categorization), a unique identifier, and the time in which the event occurred. Yet another embodiment is a device[s] that records changes in parameters that may impact the visual stimuli.